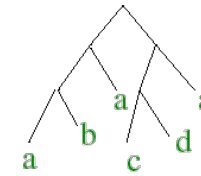
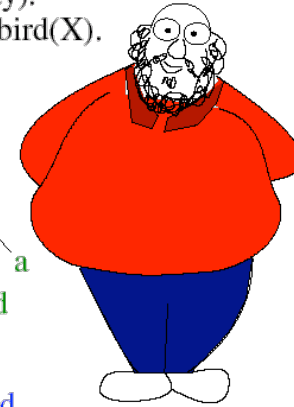


bird(tweety).
flies(X):-bird(X).



q.e.d.

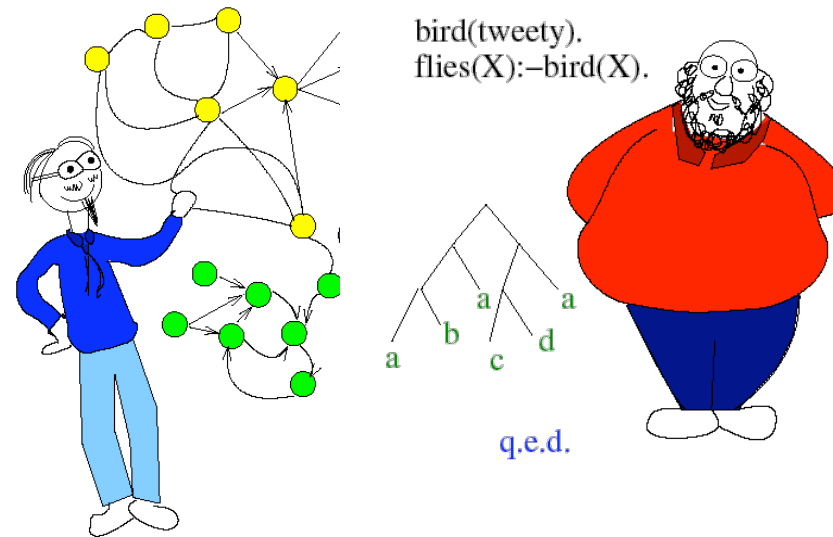


Teil 2

Teilbereiche der KI

Der intelligente Agent

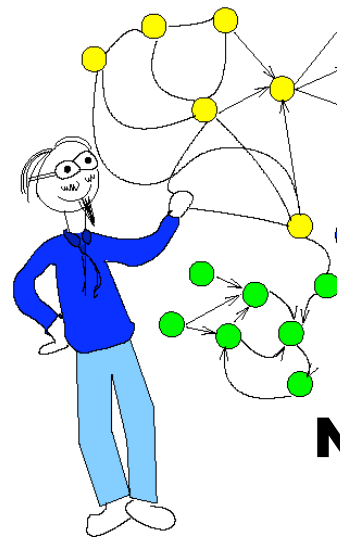
Teil 2.1



Teilbereiche der KI

Zwei Paradigmen der KI

subsymbolische KI



**Genetische
Algorithmen**

**Maschinelles
Lernen**

**Neuronale
Netze**

Planen

**Wissensreprä-
sentation**

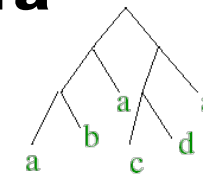
Suche

Logik

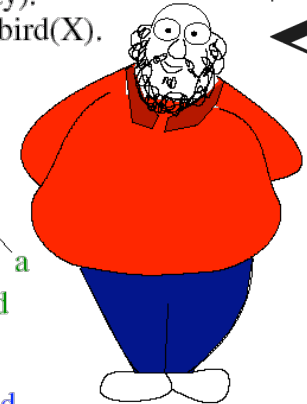
Inferenz

symbolische KI

bird(tweety).
flies(X):-bird(X).

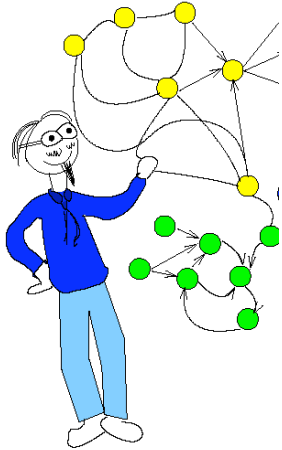


q.e.d.



- Unterscheidungsmerkmal: Ab-/Anwesenheit einer expliziten Darstellung des im System vorhandenen Wissens.
- Stärken und Schwächen sind komplementär.

Subsymbolische Verfahren



- Leistungsstarke Lernverfahren.
- Funktionieren mit verrauschten Daten.
- Robust. *Graceful degradation*.
- Häufig inspiriert von Biologie.

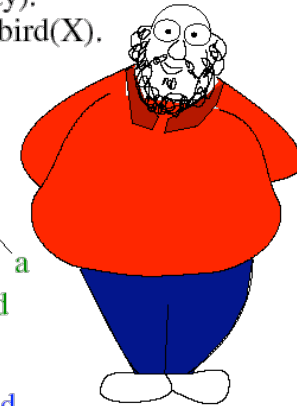
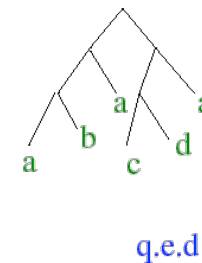
- Keine deklarative Semantik. *Black box*.
- Rekursive Strukturen schwierig.
- Verwendung von Hintergrundwissen problematisch.



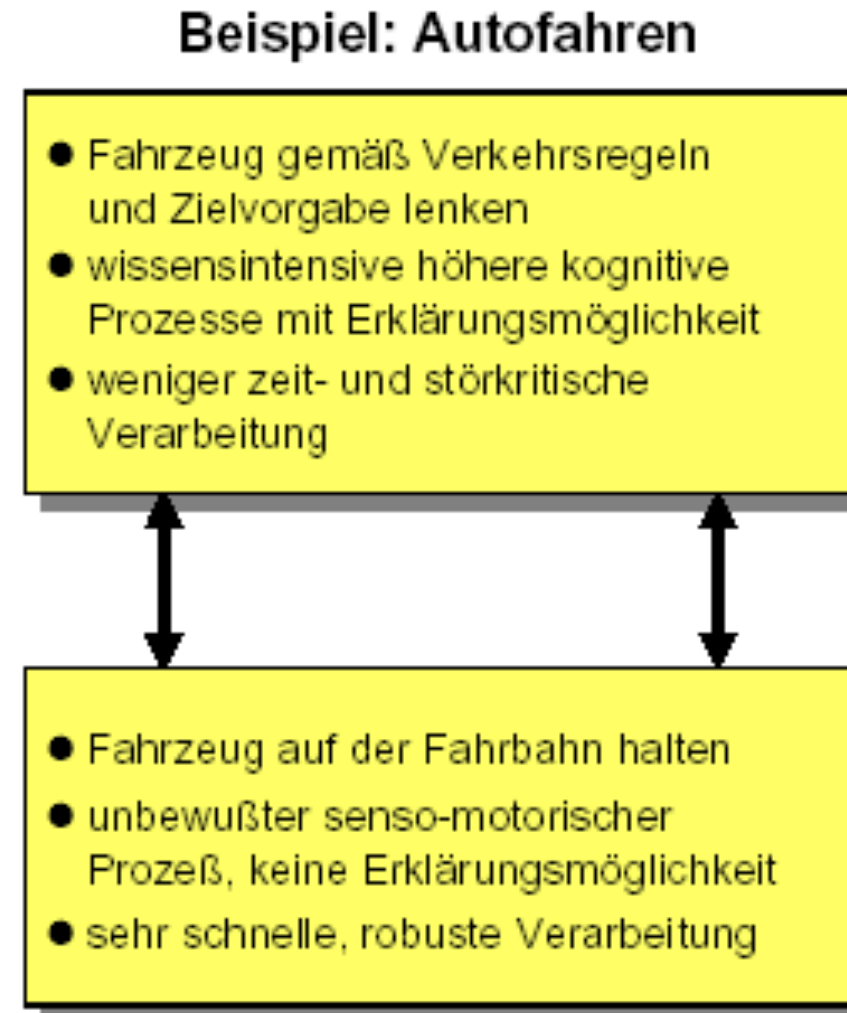
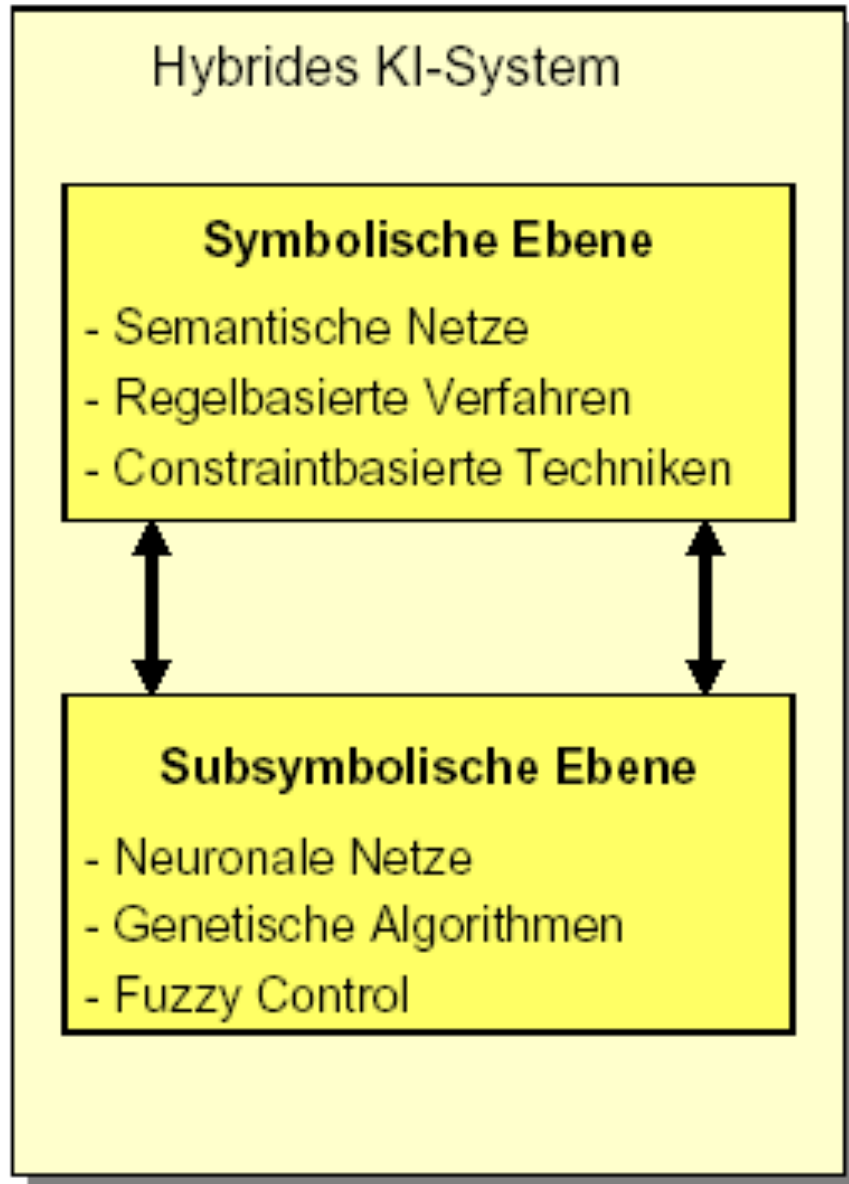
Symbolische Verfahren

- Logikbasiert. *Deklarativ*.
- Menschlichem Denken nachempfunden.
- Explizite Kodierung von Wissen. 😊
- Hochgradig rekursiv.
- Lernen schwierig.
- Empfindlich gegenüber Rauschen. ☹️
- Algorithmen haben hohe theoretische Komplexität.

```
bird(tweety).  
flies(X):-bird(X).
```

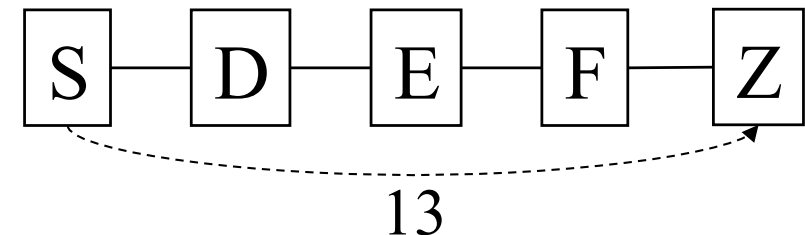
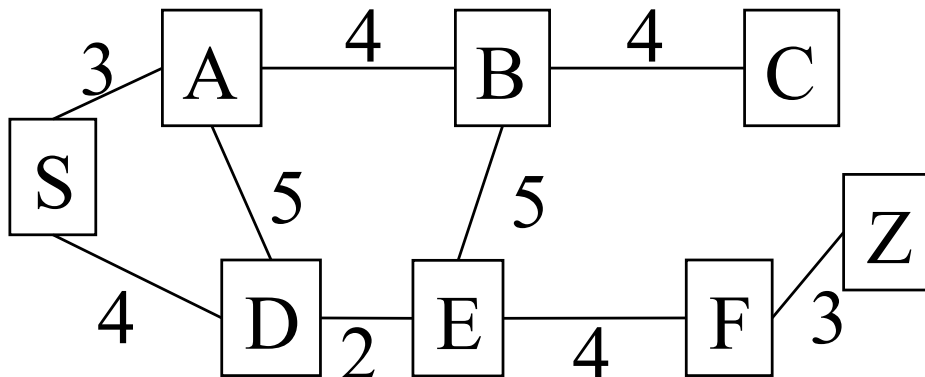


Notwendigkeit hybrider Systeme

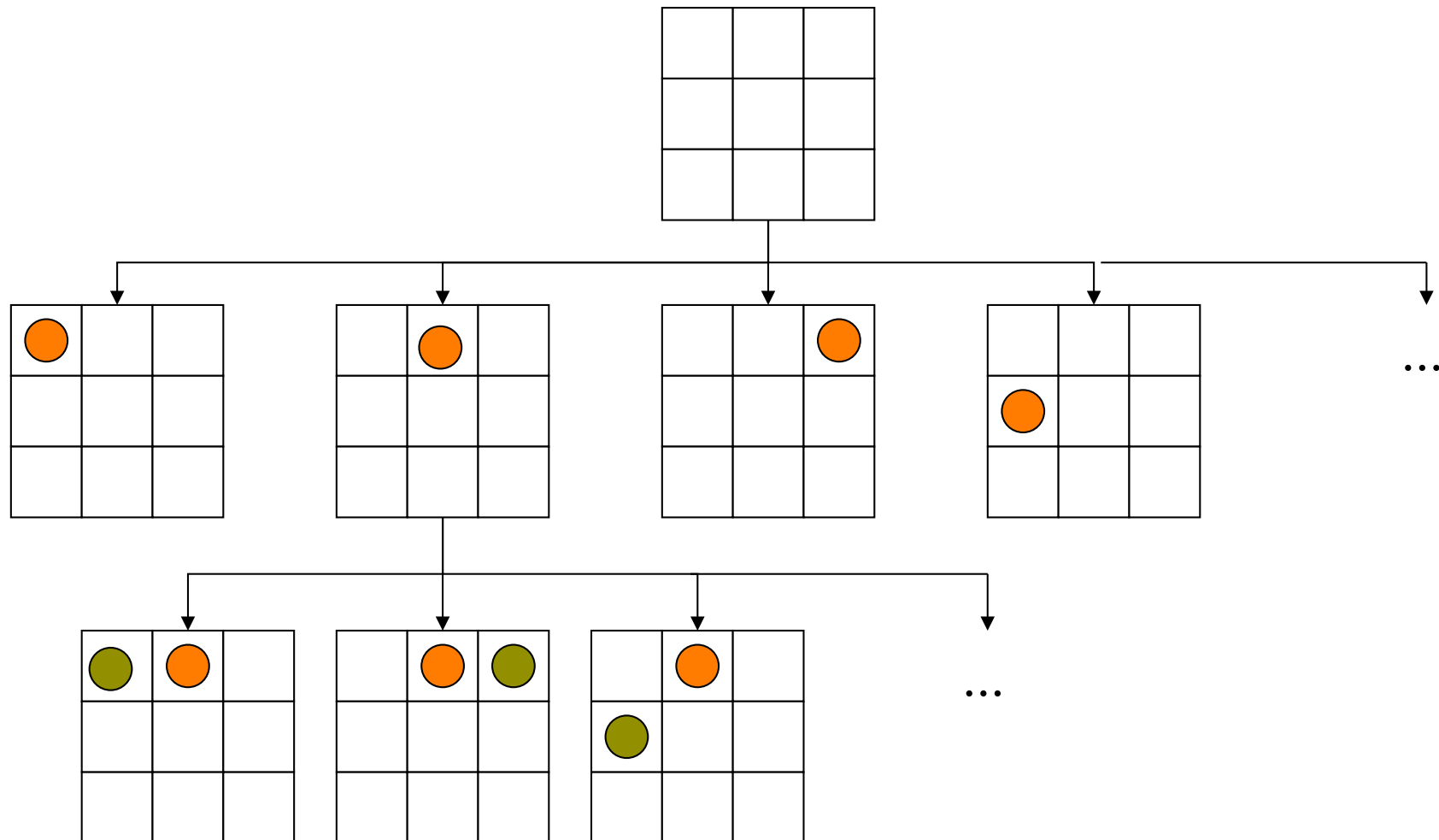


Suchverfahren

- Problemlösung durch Suche in einem Zustandsraum: gesucht wird Pfad von einem Startzustand zu einem Zielzustand
 - ⇒ nicht informierte Suche (Breiten-/Tiefensuche)
 - ⇒ informierte Suche (heuristische Suche, Optimalsuchverfahren)
 - ⇒ (un-)vollständige Kontrolle über die Zustandsübergänge (z.B. Spiele)
 - ⇒ Kostenminimierung / Gewinnmaximierung



Suche: Beispiel



● Marke Spieler A
● Marke Spieler B

Planen

- Erstellen einer Struktur (Abfolge) von Aktionen, deren Ausführung ein gegebenes Problem löst
- Unterschied zu “einfacher” Problemlösung (z.B. durch Suche):
 - ⇒ Repräsentation von Zuständen, Zielen und Aktionen meist in einen einheitlichen Repräsentationsformalismus eingebettet
 - ⇒ Suchraum riesig und strukturiert: Suchverfahren muss Struktur ausnutzen
 - ⇒ Reihenfolge bei der Erstellung des Plans nicht notwendigerweise inkrementell und vom Initialzustand ausgehend

Konfigurieren

- Syntheseprozess, bei dem Objekte aus einem vorgegebenen Anwendungsbereich unter Einhaltung bestimmter Bedingungen strukturiert zusammengefügt werden müssen.
 - ⇒ Aufgabenspezifikation bestimmt Anforderungen an die zu erzeugende Konfiguration
 - ⇒ eine Menge von Restriktionen und Relationen zwischen Objekten bestimmt Randbedingungen
- Beispiel: Rechnerkonfiguration
 - ⇒ Ziel: Funktionsfähiger Rechner, der optimiert ist für bestimmte Anwendungen (z.B. Spiele)
 - ⇒ Randbedingungen: Preis, einzelne Komponenten, CPU, Hauptspeicher, Festplatten, I/O, externe Geräte,...

Diagnose

- Suche nach einer Erklärung für ein bestimmtes (Fehl-)Verhalten in einem System
 - ⇒ heuristische Diagnose
 - ⇒ modellbasierte Diagnose
 - ⇒ fallbasierte Diagnose
 - ⇒ statistische Diagnose
 - ⇒ Diagnose mit Entscheidungsbäumen
 - ⇒ Diagnose mit Entscheidungstabellen

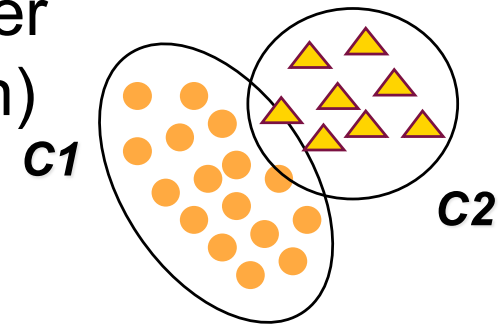
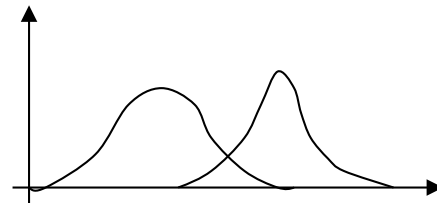
Maschinelles Lernen

- **Ziel:** (selbständige) Aneignung von Wissen und Verbesserung der Problemlösungsfähigkeit im Laufe der Zeit
- **Symbolische Lernverfahren:** Wissen ist durch Repräsentationsmechanismen mit reichhaltiger Syntax/Semantik kodiert
 - ⇒ induktives Lernen
 - ⇒ instanzenbasiertes Lernen (CBR)
 - ⇒ erklärungsbasiertes Lernen (EBL)
- **Subsymbolische Lernverfahren:** Wissen ist in einfachen Datenstrukturen wie z.B. Zahlen kodiert, Inferenz findet durch numerische oder einfache strukturelle Operationen statt
 - ⇒ genetische Algorithmen
 - ⇒ neuronale Netzwerke

Induktives Lernen

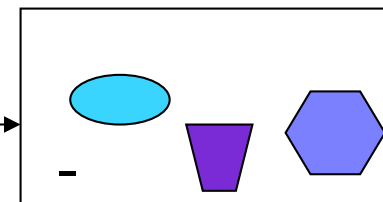
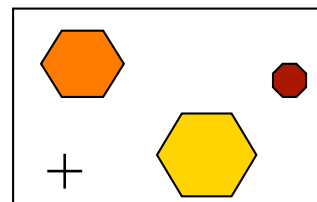
- Lernen aus Beispielen: Aufstellung einer Klassenbeschreibung durch Merkmalsanalyse vieler Beispiele aus der Klasse

⇒ numerische Beschreibung: Schätzung einer Verteilungsfunktion und deren Parameter (klassische Mustererkennungsverfahren)



⇒ symbolische Beschreibung: Ableitung einer logischen Beschreibung, verschiedene Generalisierungs- und Spezialisierungsregeln

Farbe: rötlich v gelblich
Form: 6-Eck
Größe: egal

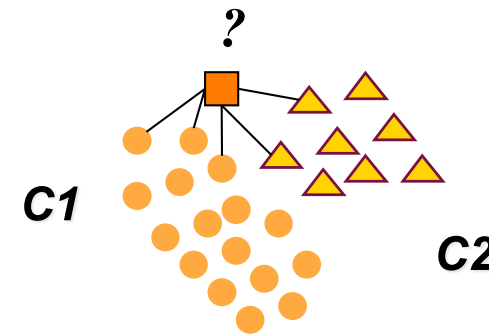
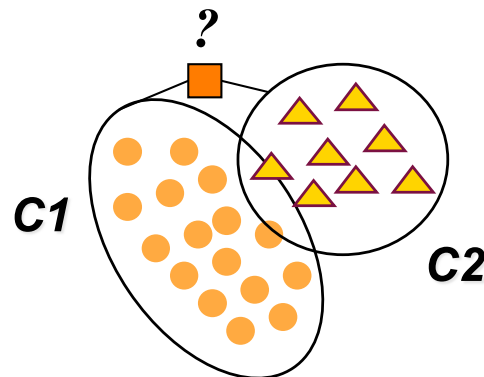


Farbe: bläulich
Form: egal
Größe: egal

Instanzenbasiertes Lernen

- Lernen aus Beispielen ohne eine Generalisierung vorzunehmen:

⇒ aus Mustererkennung: Nächster-Nachbar-Klassifikator



- ⇒ Case Based Reasoning (CBR): Um ein neues Problem zu lösen wird die bekannte Lösung eines möglichst ähnlichen Problems entsprechend angepasst
 - ✧ Ähnlichkeit von Problemen
 - ✧ Anpassung einer Lösung
 - ✧ Speicherung und Zugriff auf gelöste Probleme

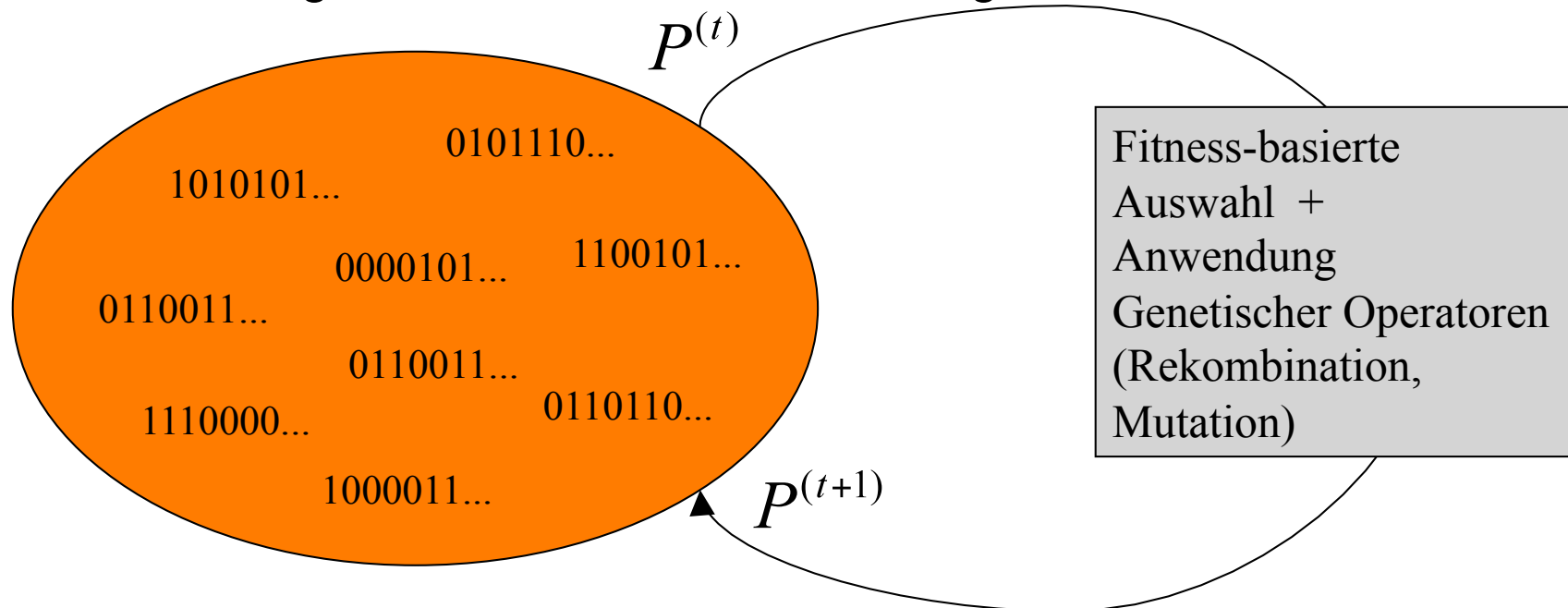
Erklärungsbasiertes Lernen

- bessere Problemlösungsfähigkeit kann auch darin bestehen, schneller zu einer Lösung zu kommen:
 - ⇒ Erklärungsbasiertes Lernen (EBL) umfasst die Umformung des Wissens in eine operationale(re) Form
 - ⇒ Modellwissen und ein einzelnes spezifisches Beispiel werden dazu verwendet, das Wissen so zu formulieren, dass es besser angewendet werden kann.

Genetische Algorithmen

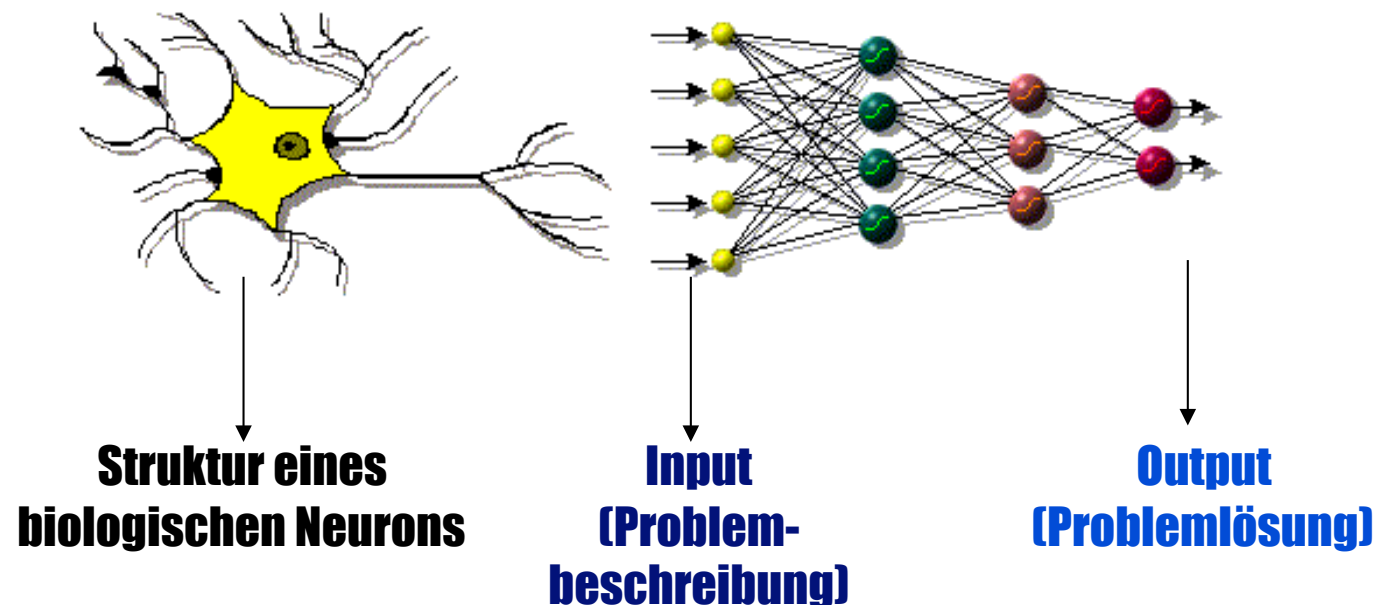
- Von John Holland in den 70er Jahren als Problemlösungsmethode entwickelt, welche sich evolutionärer Mechanismen bedient.

- ⇒ Mögliche Lösungen unterschiedlicher Qualität bilden eine Population, wobei die besten Individuen eine höhere Chance erhalten, sich fortzupflanzen
- ⇒ neue Lösungen entstehen durch Kreuzung und Mutation



Neuronale Netzwerke

- Neuronale Netzwerke sind Klassifikatoren, deren Arbeitsweise bestimmte Charakteristika des menschlichen neuronalen Systems imitiert
 - ⇒ Menge von selbstständig rechnenden Einheiten, die Neuronen genannt werden.



Data Mining

- Auswertung großer, unübersichtlicher Datenmengen mit dem Ziel der Entdeckung verborgener allgemeiner Muster
 - ⇒ Beziehungen zwischen Objekten
 - ⇒ konzeptuelle oder numerische Regelmäßigkeiten oder Anomalien
- statistische Methoden und maschinelle Lernverfahren
- eingesetzt z.B. bei Data Warehouses

Wissensrepräsentation

- formale Darstellung von Wissen über einen bestimmten Bereich (Modellierung von Objekten, Beziehungen, Ereignissen, ...)
- Begrenztes Abbild der Realität
 - ⇒ explizite Repräsentation
 - ✧ Aussagen, Prädikate
 - ✧ Frames, semantische Netzwerke
 - ✧ Regeln, Produktionssysteme
 - ⇒ implizite Repräsentation
 - ✧ Programme
- Ohne Operatoren zur Verarbeitung des Wissens wertlos.
 - ⇒ Inferenzmechanismen

Wissensbasierte Systeme

- Trennung von Wissensrepräsentation und Operatoren zur Anwendung des Wissens
 - ⇒ explizite Wissensrepräsentation
 - ⇒ Problemlösungskomponente ist unabhängig vom Anwendungsbereich
 - ⇒ Wissen über einen Bereich kann unabhängig vom Rest des Systems (Operatoren zur Anwendung des Wissens) bearbeitet bzw. ausgetauscht werden
- Expertensystem:
 - ⇒ Modellierung des Wissens über einen bestimmten Anwendungsbereich zur Lösung spezifischer Probleme aus diesem Bereich
 - ⇒ System repräsentiert menschlichen Experten für einen bestimmten Bereich

Inferenz

- Schlussfolgerungsmechanismen, mit deren Hilfe aus bekanntem Wissen “neues” Wissen abgeleitet werden kann.
- Logische Inferenz
 - ⇒ Anwendung von Ableitungsregeln
 - ✧ Deduktion: folgerichtiges Schließen
 - ✧ Induktion: generalisierendes Schließen
 - ✧ Abduktion: ursächliches Schließen
 - ⇒ Annahmen über die Welt, z.B. Closed World Assumption, (nicht) monotone Logiken
 - ⇒ Unsicherheiten (probabilistische oder unscharfe Logiken)
- Heuristische Inferenz
 - ⇒ “Daumenregeln”

Logische Kalküle I

- Logik/logischer Kalkül:
 - ⇒ formales System zur Beschreibung des Wissens (Repräsentationsformalismus)
 - ⇒ Menge von Ableitungsregeln zur Herleitung logischer Folgerungen (Inferenz)
- Aussagenlogik
 - ⇒ Aussagen, einfache logische Verknüpfungen wie z.B. \wedge , \vee , \neg , Resolution, Modus Ponens, ...
- Prädikatenlogik
 - ⇒ Prädikate, logische Verknüpfungen, Variablen und deren Quantifizierung (\forall, \exists), Resolution...

Logische Kalküle II

➤ Sortenlogik:

- ⇒ Einführung von “Typen”, die bei der Unifikation eine Rolle spielen

➤ Modallogik(en):

- ⇒ Quantoren werden durch Modalitäten (z.B. \Box , \Diamond) ergänzt, mit denen Möglichkeiten modelliert werden können (notwendig / möglich) und
- ⇒ die je nach Anwendungsgebiet unterschiedlich interpretiert werden, z.B. in Temporallogik: irgendwann / immer

Logische Kalküle III

➤ Nichtmonotone Logiken

- ⇒ Möglichkeit der Invalidierung von Schlussfolgerungen durch hinzukommendes neues Wissen
- ⇒ z.B. closed-world reasoning: jede nicht explizit gegebene Grundaussage wird als falsch angenommen
- ⇒ z.B. Ermangelungsschließen: unvollständiges Wissen wird durch Standardannahmen ergänzt (z.B. “A’s sind typischerweise B’s”)

Logische Kalküle IV

➤ Probabilistische Logik

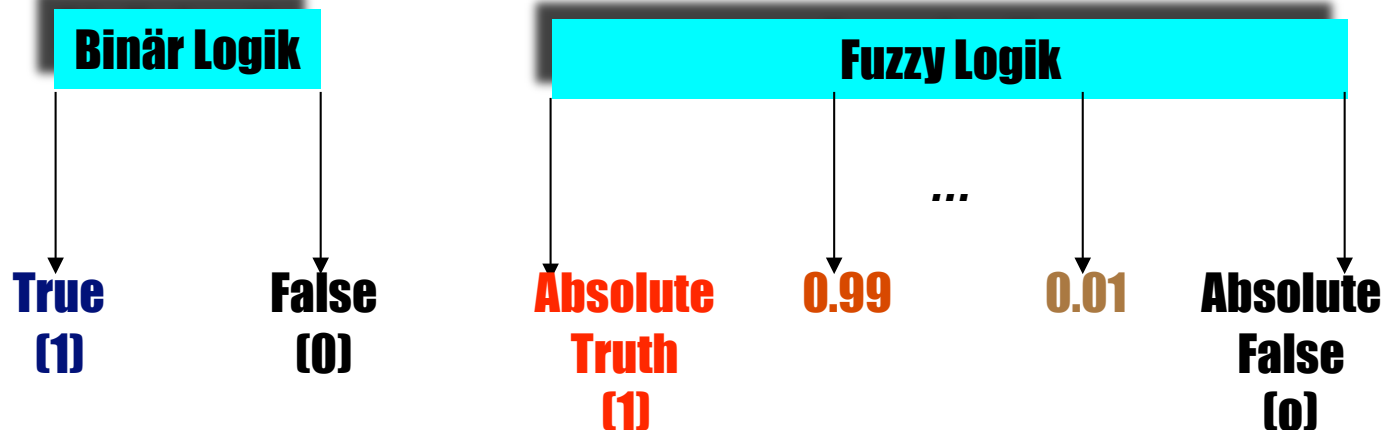
- ⇒ Problem: Entscheidungen müssen getroffen werden, ohne dass alles Wissen, das dazu erforderlich ist, bekannt ist.
- ⇒ nicht mehr exakt logisch, sondern Berücksichtigung von Unsicherheitsfaktoren für Fakten und Schlussfolgerungen

➤ Fuzzy Logic

- ⇒ unscharfes Schließen, nicht nur “ja”/”nein”, sondern auch “eher ja” / “eher nein” (verschiedene Abstufungen von Gültigkeit)

Fuzzy Logic

- Entwickelt von Lotfi Zadeh in den 60er Jahren, mit dem Ziel, bestimmte Probleme aus der klassischen Logik (Binär- Logik) zu lösen
- Fuzzy Logic umfasst 0 and 1 als Extremfälle eines Wahrheitswerts, aber ebenso Zwischenwerte
 - ⇒ ähnlicher zu dem, wie ein Mensch denkt





Teil 2.2

Der intelligente Agent

Aktueller Trend: Agent

- Bisher sind noch viele KI-Systeme
 - ⇒ spezifisch auf einen Zweck ausgerichtet und
 - ⇒ berücksichtigen zumeist nur einen oder zumindest nur wenige Aspekte menschlicher Intelligenz.
 - ⇒ Nicht eingebettet in Umgebung -> kein **Embodiment**
- Angestrebt werden Systeme, die umfassender sind bzw. mehr Aspekte abdecken:
 - ⇒ **weniger**: riesige Systeme wie z.B. CYC (Wissensbasis mit mehr als 1 Mio. Regeln), die einen breiten Einsatz anstreben.
 - ⇒ **mehr**: verteilte Intelligenz mit Hilfe von Agenten, die bei Bedarf miteinander kooperieren

Rationaler Agent

- Zentrale Sichtweise im folgenden die eines Agenten:
 - ⇒ Ein Agent ist eine autonome Einheit, welche ihre Umwelt wahrnimmt und in ihr agiert.
 - ⇒ Ein rationaler Agent richtet seine Handlungen so aus, dass ein vorgegebenes Ziel möglichst gut erreicht wird.
 - ⇒ vereinfachte Formale Beschreibung durch Funktion

$$f : P^* \mapsto A$$

- ⇒ Gesucht wird jeweils ein Agent, der die beste Performance hat (unter Wahrung bestimmter Restriktionen)

Rationalität

➤ Rationalität bedeutet nicht

- ⇒ Allwissenheit
- ⇒ Hellsichtigkeit
- ⇒ garantierter Erfolg

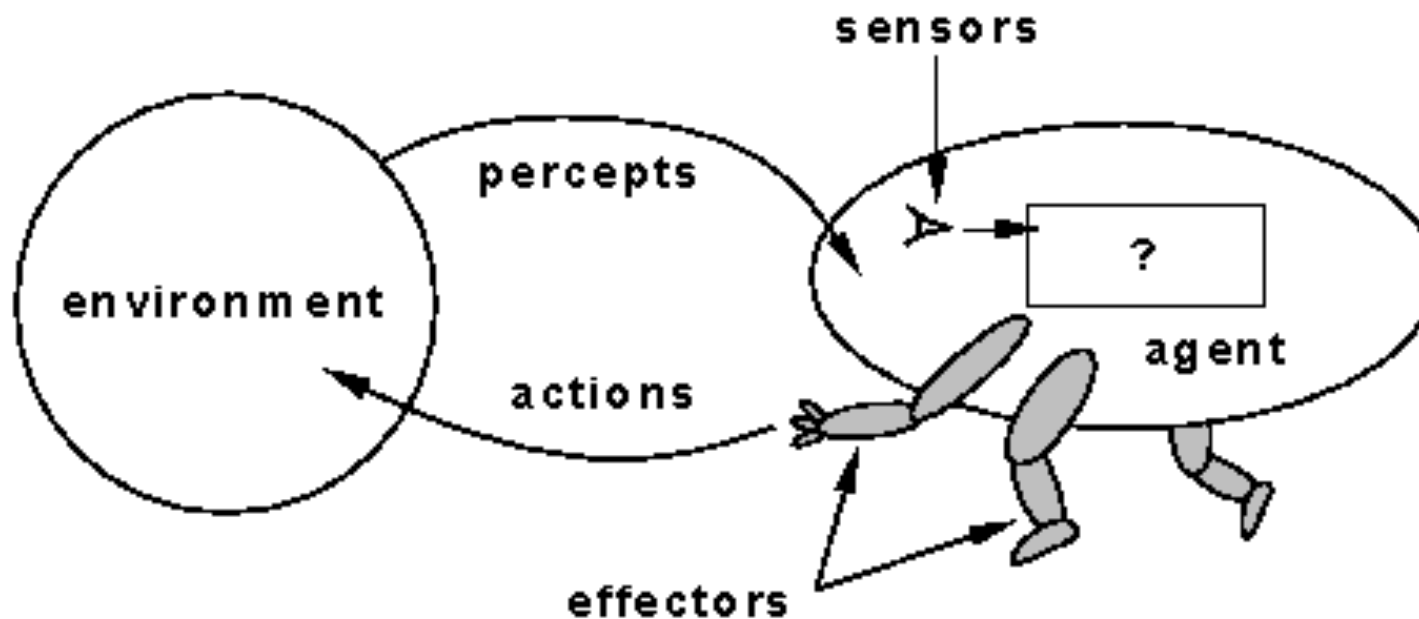
sondern bezieht sich auf den erwarteten (erwartbaren) Erfolg

➤ Idealer rationaler Agent versucht seine Performance zu optimieren, indem er unter Berücksichtigung der Beobachtungen aus der Umwelt und seines Wissens über dieser Umwelt entsprechende Aktionen ausführt.

- ⇒ Performance-Maß
- ⇒ Wahrnehmungs-Sequenzen aus der Umwelt
- ⇒ Wissen über die Umgebung
- ⇒ durchführbare Aktionen

Rationaler Agent

- P** Percepts (Wahrnehmung)
- A** Actions (Aktionen)
- G** Goals (Ziele)
- E** Environment (Umgebung)



Beispiel

Automatisierter Taxifahrer

➤ Wahrnehmungen (Percepts)

⇒ Video, Beschleunigungsmesser, GPS, ...

➤ Aktionen (Actions)

⇒ steuern, beschleunigen, bremsen, sprechen, ...

➤ Ziele (Goals)

⇒ Ort erreichen, Profit maximieren, Sicherheit, Verkehrsregeln beachten, ...

➤ Umgebung (Environment)

⇒ Straßen einer Stadt, Autobahn, Fußgänger, Radfahrer, Wetter, ...

Anforderungen I

- Agent ist definiert in einer Umgebung und besitzt Wissen über sie:
 - ⇒ Frage nach Wissensrepräsentation
- Agent empfängt Nachrichten aus seiner Umwelt und handelt entsprechend
 - ⇒ Frage nach Wissensverarbeitung
- Agent besitzt Ziele, die er erfüllen will
 - ⇒ Frage nach Problemlösung (Wie kann das Ziel erreicht werden?)
 - ⇒ Frage der Planung von Aktionen
- Agent will seine Ziele möglichst gut erfüllen
 - ⇒ Frage nach Verbesserung durch Lernen

Anforderungen II

- Agent muss evtl. in einer menschlich geprägten Umgebung handeln
 - ⇒ Frage nach Interpretation von Sensordaten (Bildanalyse)
 - ⇒ Frage nach Navigation und Handhabung (Robotik)
 - ⇒ Frage nach Kommunikation mit Menschen (NLP)
- Agent muss evtl. herausfinden, warum etwas nicht wie geplant funktioniert hat
 - ⇒ Frage nach (Fehler-) Diagnose

Umgebungs-Eigenschaften

➤ Zugänglichkeit:

- ⇒ Sind alle für eine Entscheidung notwendigen Teile der Umgebung beobachtbar?
- ⇒ zugänglich / nicht zugänglich

➤ Determiniertheit:

- ⇒ Bestimmt der aktuelle Zustand der Umgebung zusammen mit der gewählten Aktion eindeutig den Folgezustand?
- ⇒ deterministisch / nicht deterministisch

Umgebungs-Eigenschaften

➤ **Episodenhaftigkeit:**

- ⇒ Kann die Erfahrung des Agenten in einzelne Episoden (Wahrnehmung + Handlung, keine Historie) eingeteilt werden
- ⇒ episodenhaft / nicht episodenhaft

➤ **Veränderlichkeit:**

- ⇒ Kann sich die Umgebung während des Entscheidungsfindungsprozesses verändern?
- ⇒ statisch / dynamisch

➤ **Auflösung:**

- ⇒ Ist der Raum der Wahrnehmungen und Aktionen diskret oder kontinuierlich?
- ⇒ diskret / kontinuierlich

Beispiele

Umgebung	zugänglich	deterministisch	episodisch	statisch	diskret
Schach	ja	ja	nein	teilw.	ja
Poker	nein	nein	nein	ja	ja
Backgammon	ja	nein	nein	ja	ja
Taxi	nein	nein	nein	nein	nein
medizinisches Diagnosesystem	nein	nein	nein	nein	nein
Bildanalysesystem	ja	ja	ja	teilw.	ja
reale Welt	nein	nein	nein	nein	nein

Struktur eines Agenten

- Agent = Architektur + Programm
 - ⇒ **Architektur:** Aufbau der Ein- und Ausgabegeräte sowie der Verarbeitungseinheiten des Agenten (Sensoren, Effektoren, Prozessor)
 - ⇒ **Programm:** Funktion, die die Abbildung der Wahrnehmungen auf Aktionen vornimmt

$$f_W : P \mapsto A$$

```

function SKELETON-AGENT(percept) returns action
static: memory // the agent's memory of the world
memory      ← UPDATE-MEMORY(memory, percept)
action      ← CHOOSE-BEST-ACTION(memory)
memory      ← UPDATE-MEMORY(memory, action)
return action
  
```

Lookup Agent

```

function TABLE-DRIVEN-AGENT(percept) returns action
static: percepts    // a sequence, initially empty
           table,      // a table, indexed by percept sequences,
                       // initially fully specified
append percept to the end of percepts
action      ← LOOKUP(percepts, table)
return action

```

➤ Problem: Vollständige Abbildung ist oft

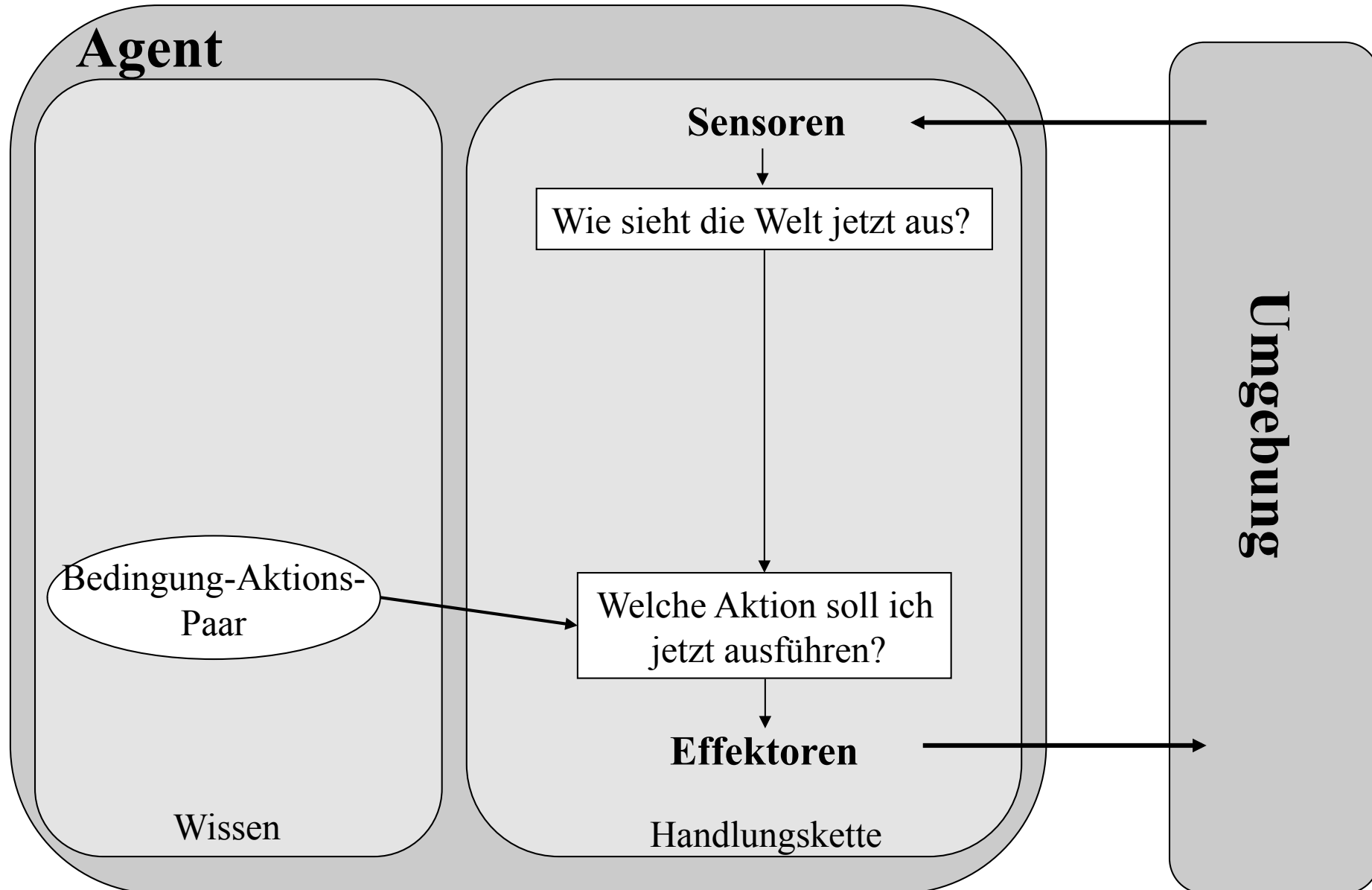
Wahrnehmung x	Aktion z
x_1	z_1
x_2	z_2
...	...

- ⇒ nicht möglich (z.B. wenn x nicht diskret) oder
- ⇒ zu aufwendig (z.B. Schach: ca. 10^{43} Positionen) und Lernen dauert dementsprechend zu lange
- ⇒ Wissen ist zu statisch: Änderungen der Umwelt führen zu Fehlverhalten

Typen von Agenten

- Je nachdem, wie die Abbildung durchgeführt wird (Nutzung von Wissen), unterschiedliche Typen:
 - ⇒ **einfacher, reagierender Agent**: nutzt einfach Abbildung von Wahrnehmungen zu Aktionen
 - ⇒ **reagierender Agent mit internen Zuständen**: arbeitet mit Gedächtnis um den aktuellen Zustand der Welt besser zu bestimmen
 - ⇒ **zielorientierter Agent**: verwendet Wissen über wünschenswerte Zustände (zu erreichende Ziele), entscheidet, ob eine Aktion diesbezüglich sinnvoll ist
 - ⇒ **nützlichkeitsbasierter Agent**: gewichtet zusätzlich Zustände, um den Wert unterschiedlicher Aktionen zu beurteilen

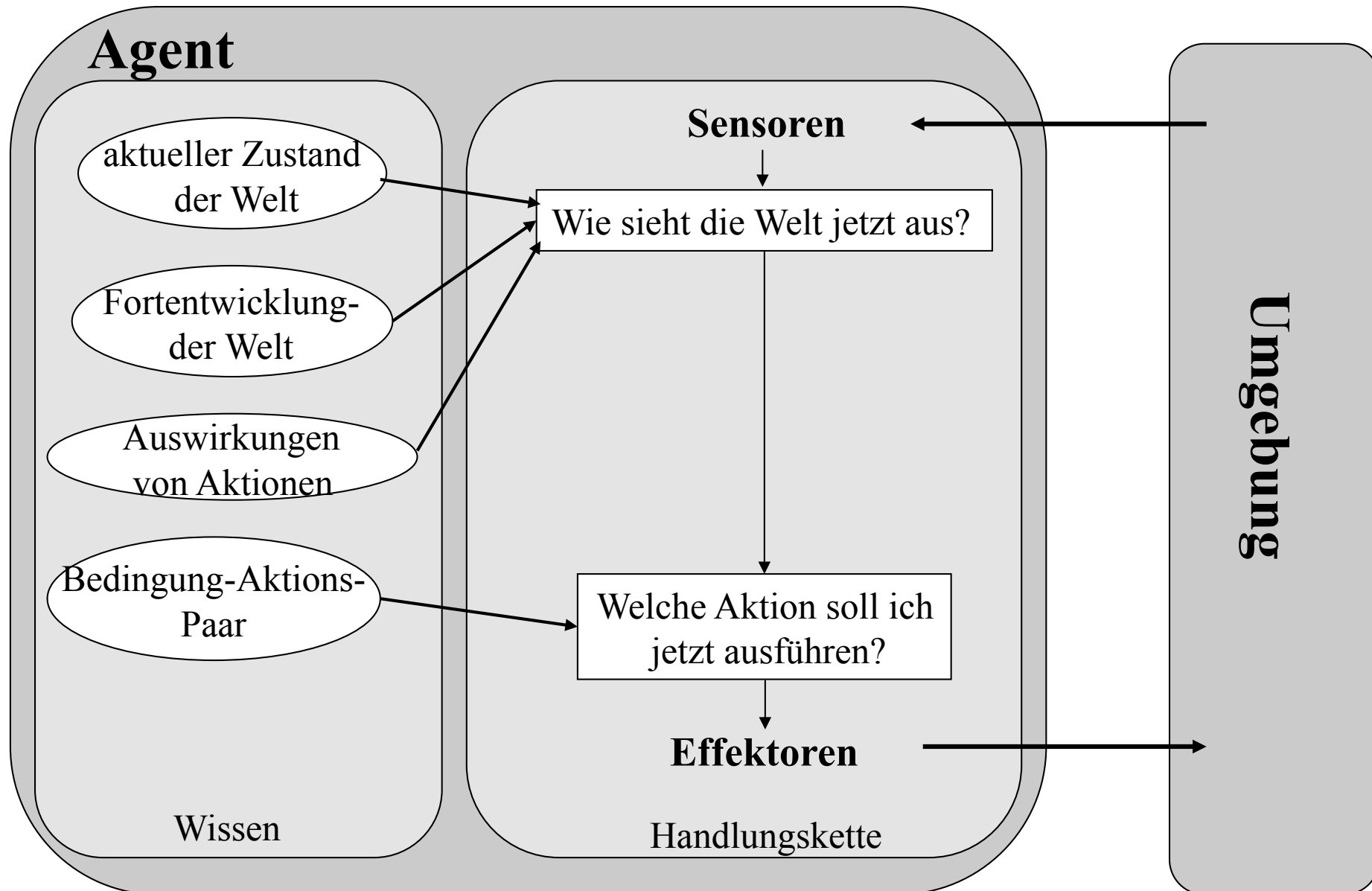
Reagierender Agent



Reagierender Agent

```
function SIMPLE-REFLEX-AGENT(percept) returns action  
static: rules // a set of condition-action rules  
state          ← INTERPRET-INPUT(percept)  
rule          ← RULE-MATCH(state, rules)  
action        ← RULE-ACTION[rule]  
return action
```

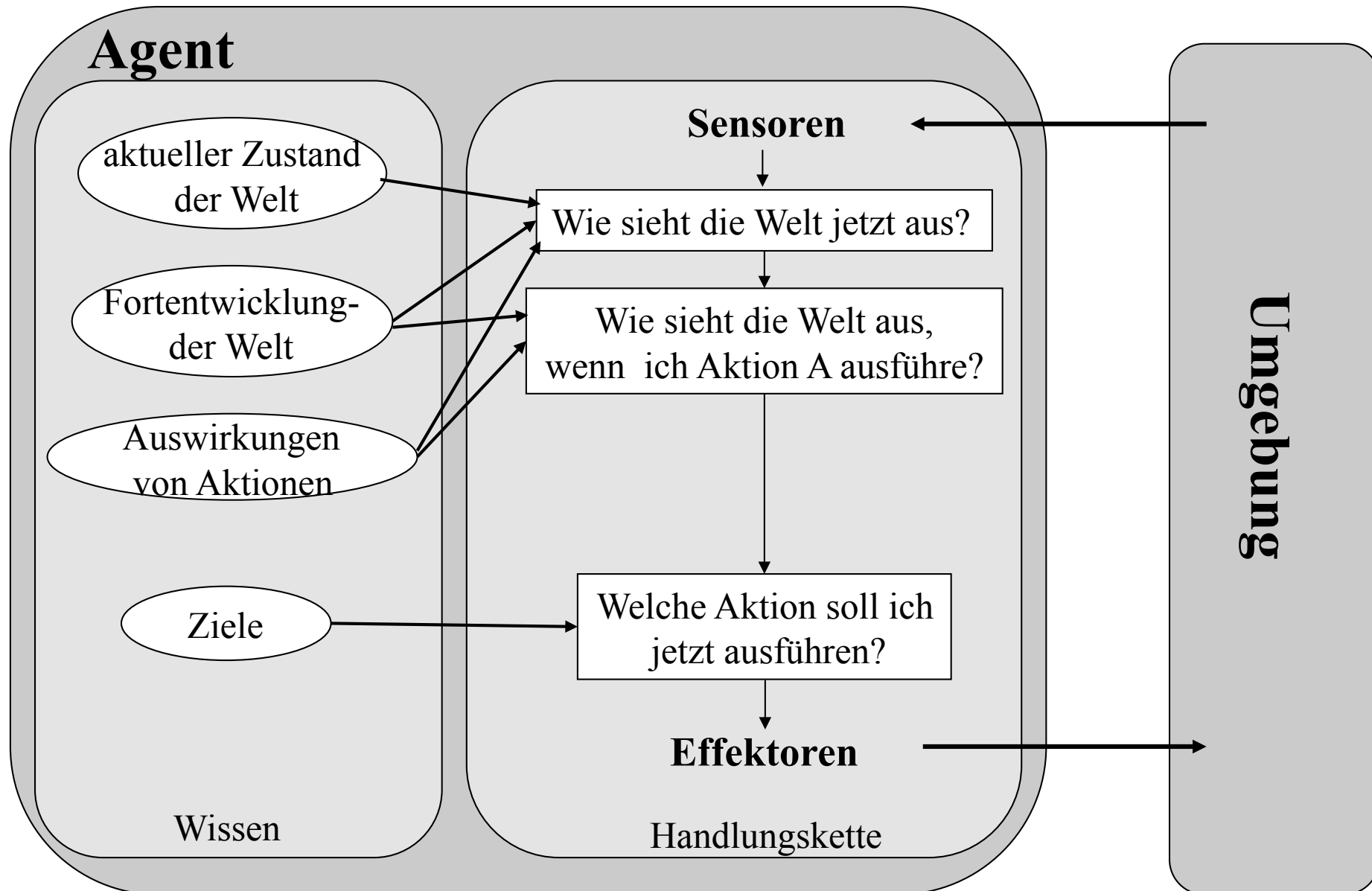
Reagierender Agent mit internen Zuständen



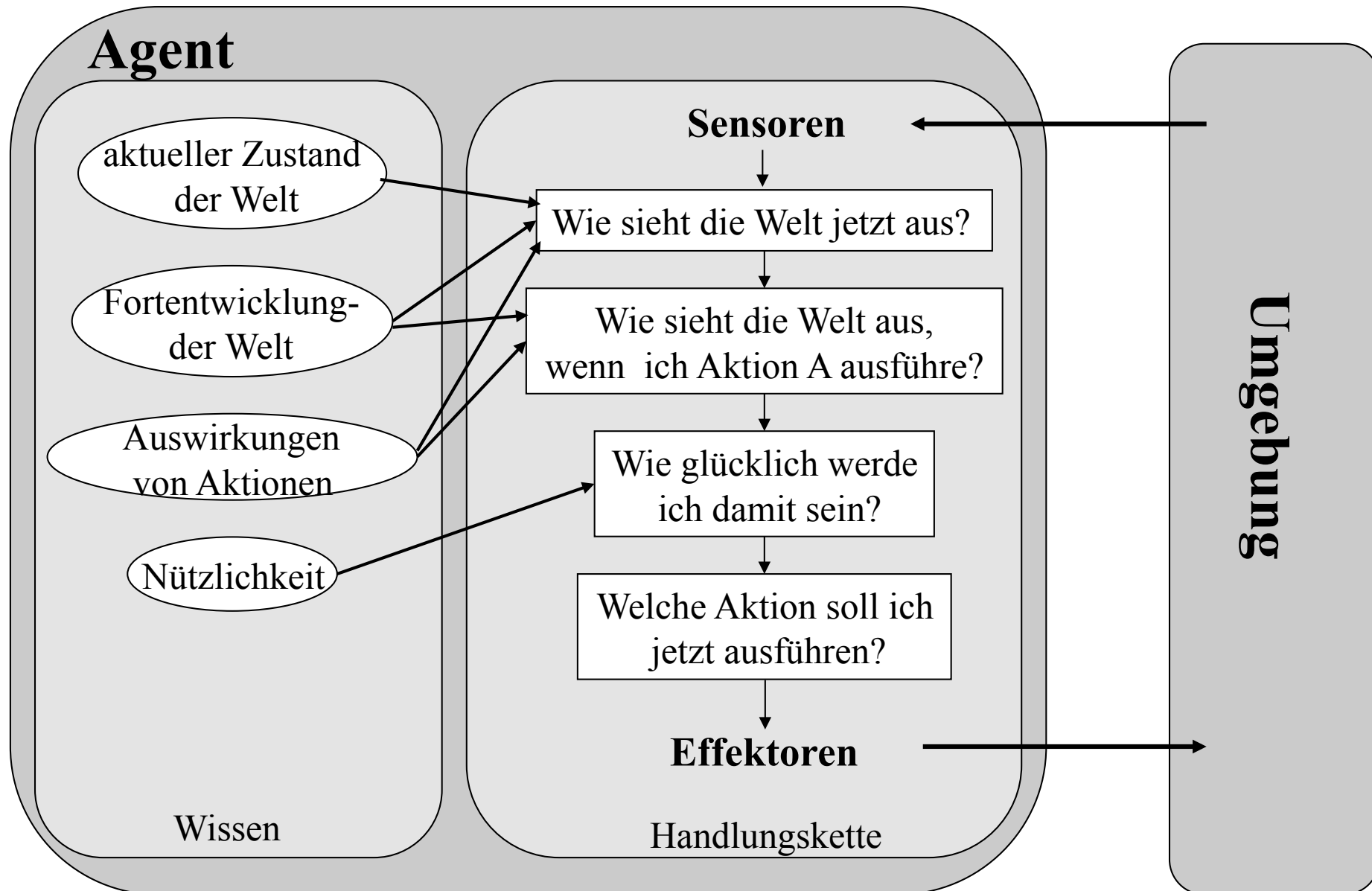
Reagierender Agent mit internen Zuständen

```
function REFLEX-AGENT-WITH-STATE(percept) returns action  
static: state // a description of the current world state  
         rules // a set of condition-action rules  
state      ← UPDATE-STATE(state, percept)  
rule       ← RULE-MATCH(state, rules)  
action     ← RULE-ACTION[rule]  
state      ← UPDATE-STATE(state, action)  
return action
```

Zielorientierter Agent



Nützlichkeitsbasierter Agent



Ausblick: Multiagentensysteme

➤ Agent design:

⇒ Wie konstruieren wir Agenten, die unabhängig und autonom agieren, um die an sie delegierten Aufgaben zu erledigen?

➤ Society design:

⇒ Wie konstruieren wir Agenten, die mit anderen Agenten interagieren, um ihre Aufgaben zu erfüllen, auch wenn manche dieser Agenten gegensätzliche Interessen haben und ihre eigenen (konkurrierenden) Ziele verfolgen?

Wirkmechanismen in MAS

- keine zentrale Steuerung
- *bottom up*: spezifische Fähigkeiten der Agenten führen zum gewünschten Interaktionsverhalten auf Gruppenebene (s.a. emergent behaviour).
- *top down*: spezifische Regeln auf Gruppenebene (wie Konventionen, Normen) bestimmen das Interaktionsrepertoire auf der Ebene einzelner Agenten.

Neue Aspekte bei MAS

- Kommunikation, Verhandlung und Vereinbarung: Sprachen und Protokolle
- Repräsentation von und Schlussfolgern über Wissen, Aktionen, und Pläne anderer Agenten sowie Interaktionsprozesse
- Vermeidung chaotischen Systemverhaltens
- Bildung und Auflösung von Organisationsstrukturen

Robo-Cup



By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.

<http://www.robocup2004.pt/photosAndVideos/videos/humanoid>



nächste Vorlesung:

Suchverfahren